IBM® DB2® Data Links Manager Performance Evaluation Results

August 12, 2002

Executive summary

This paper presents a study of DB2 Universal Database™ Data Links performance. Data Links provides a powerful data control system, including referential integrity, access control, recovery capabilities, and transaction consistency.

By using the file system paradigm, system administration costs and complexities are reduced. Data Links offers a flexible system for e-commerce and Internet applications. Measurements are based on most-common user operations, and in order to provide results in realistic conditions, the standard database from the Transaction Processing Performance Council Benchmark Web™ (TPC-W™) will be used.

To demonstrate the advantages of Data Links, tests compare Data Links with its BLOB (binary large object) alternative. Details concerning methodology and configurations used are described, followed by comments on the results.

For more informations about Data Links, refer to :
http://www-3.ibm.com/software/data/db2/datalinks/

Contents

Test environment

All measurements were run on AIX® 4.3.3 (32 bit) machines with DB2 Universal Database™ Version 7.2, connected by a high speed (10 Mbps) LAN.
The machine names[1] and features are as follows:

- **Nike** : IBM 7013 J30, 8 processors, 1 GB memory , 3 SSA 36 GB Disks, used for Data Links files, archive files, BLOBs. 6 disks 4.5 MB SCSI, used for home directories. Also used for the single machine configuration.

- **Apollo** : IBM 7248 C02, 1 processor, 196 MB memory, 2 SCSI disks (1 GB and 2 GB). DB2 server for multiple machine Data Links configuration.

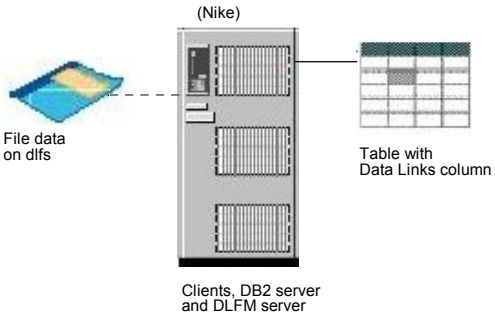
- **Achilles** : IBM 7248 C02, 1 processor, 131 MB memory, 2 SCSI disks (1 GB and 2 GB). Client machine for multiple machine configuration.

For Data Links and BLOBs, two configurations are used : Multiple machine and single machine. The single machine configuration allows us to test without network and NFS activities.

Data Links configuration

The system using DB2 Data Links technology includes 3 components : Data Links server, DB2 server, and DB2 client. The DLFM (Data Links File Manager) server hosts the DLFS (Data Links File System) containing files under Data Links control.

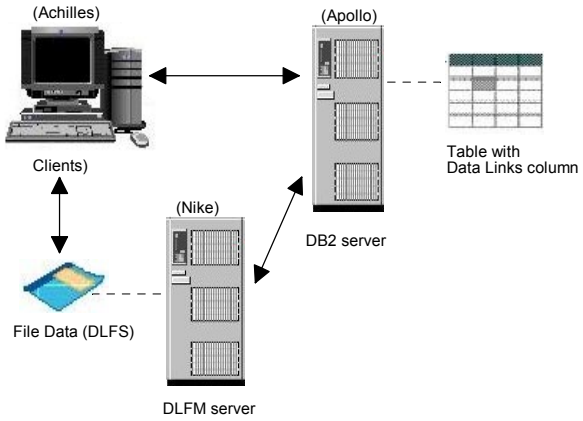
Figure 1. *Single machine*



In this example, all components are installed on the same machine using different user ids. Measurements are performed with the DB2 user id. The database, the DLFS, and the archive are located on different SSA logical drive disks.

[1] The machines names Nike, Apollo and Achilles are derived from the names of ancient Greek gods and do not refer to any modern products or companies.

Figure 2 . *Multiple machines*

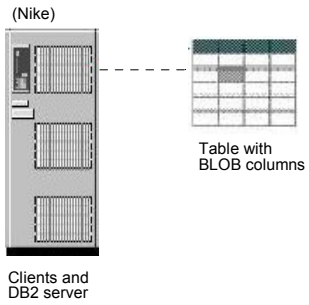


Each component is installed on a separate machine. The client launches all tests in this configuration. Nike computer is used as DLFM server. The database on the Apollo computer is located on a local SCSI disk drive. The DLFS is mounted with NFS3 on Achilles and Apollo computer. (The TCP/IP communications protocol is used to access the remote node). Measurements are performed on the DB2 client (Achilles computer).

Large object (LOB) configuration

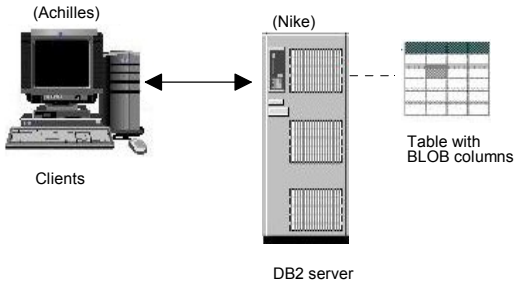
The system includes a DB2 server and a DB2 client. Performance tests will use binary large objects (BLOBs) .

Figure 3. *Single machine*



In this example, data is inserted directly into the database. In order to have a fair comparison between Data Links and BLOBs, the database is located on the same device type as the DLFS (SSA).

Figure 4 . *Multiple machines*



The TCP/IP communications protocol is used to communicate between the remote client and Nike. The database on the DB2 server is located on the same disk as in the single machine configuration. Contrary to the Data Links configuration, the DB2 server here is located on Nike. However, test data are on Nike for both configurations.

Layout of the database

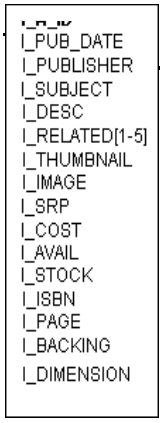
All tests are based on the TPC-W™ database[2]. The TPC-W database comes from the TPC-W benchmark. This is a transactional Web e-commerce benchmark. It is a standard unit of measurement to compare systems and solutions.

TPC-W database

This database contains an item table that is used to compare scenarios. The TPC-W database provides a full database schema with a customer table, orders table, address table, item table, and soon, all with indexes. Only the item table is used and changed to perform tests. All measurements are based on object management related to the contents of the item table.

The structure of this table follows:

Figure 5. *Item table*



The I_THUMBNAIL and I_IMAGE columns are used to compare different type columns (DATALINK versus BLOB).

The standard distribution for the different file sizes follows:
Images: 5 Kb (45%), 10 Kb (35%), 50 Kb (15%), 100 Kb (4%), and 250 Kb (1%). Weighted average is **19.7 Kb**.
Thumbnails: 5 Kb

This distribution represents an e-commerce for Web application, and the most common files for many Data Links customers are generally quite large. Therefore an additional distribution has been created for these tests. This database is called the big files database.

Big files database

The database structure is identical to the TPC-W database. Only the object size distribution is different. The big files database distribution follows:
Images: 1 MB (80%), 10 MB (20%). Weighted average is **2.8 MB**.
Thumbnails: Not used (fixed as varchar)

Note: The TPC-W database includes **2 columns** (Image and Thumbnail) whose contents were tested, but the big files database includes only one (Image). The purpose of the big files database is to compare big files, therefore thumbnails are not very useful.

These 2 databases give an approximation of 2 different database system environments. Where appropriate, additional tests have also been done on files of more widely varying sizes. In this case, only the Image column is used; these results are represented with line-graph charts. This distribution has been created with a large-scale size from 5 Kb to 5 Mb.

[2] For additional background information about TPC-W, refer to: http://www.tpc.org/tpcw/default.asp

Test plan

The following is the list of tests performed. The exact test scenario (language, options, etc.) for each case and the choice of configurations are explained along with the results. Analyses are included and the system state is described. The number of clients has been chosen to give meaningful results. For each result, the choice of number of clients is also explained.

1. Average time to access a file on disk

Tests	Configurations
Comparison of Data Links / BLOBs with a mix of different file sizes	• TPC-W database / big files database • single machine / multiple machines • 1 client / multiple clients
Comparison of Data Links / BLOBs with individually different file sizes	• 1 client • single machine / multiple machines
Comparison of Data Links Read permission DB / Read permission FS	• TPC-W database • single machine • 1 client / multiple clients
Comparison of Data Links MAC0 / MAC1	• TPC-W database • single machine • 1 client / multiple clients

2. Average time to load a file into memory

Tests	Configurations
Comparison of Data Links / BLOBs with a mix of different file sizes.	• TPC-W database / big files database • single machine / multiple machines • 1 client / multiple clients
Comparison of Data Links Read permission DB / Read permission FS / Blobs with individually different file sizes	• 1 client • single machine / multiple machines

3. Average time to insert/update/delete a file

Tests	Configurations
Comparison of Data Links / BLOBs with a mix of different file sizes.	• TPC-W database / big files database • single machine / multiple machines • 1 client / multiple clients
Comparison of Data Links / BLOBs with individually different file sizes	• 1 client • single machine / multiple machines
Comparison of Data Links Read permission DB / Read permission FS	• TPC-W database • single machine • 1 client / multiple clients
Comparison of Data Links Recovery Yes / Recovery No	• TPC-W database • single machine • 1 client / multiple clients

4. Average time to back up

Tests	Configurations
Comparison of Data Links / BLOBs for different databases	• TPC-W database / big files database • single machine • 1 client

5. Average time to load data

Tests	Configurations
Comparison of Data Links for different configurations	• TPC-W database / big files database • 1 client • Load utility / Import utility • single machine / multiple machines
Comparison with BLOBs	• TPC-W database • single machine • 1 client / multiple clients
Comparison of Data Links Recovery Yes / Recovery No	• TPC-W database • single machine • 1 client / multiple clients

6. Average time to reconcile

Tests	Configurations
Comparison of Data Links for different configurations	• TPC-W database / big files database • single machine • 1 client • Fix-up required / No fix-up required

DB2 Versions

Tests	Versions
Time to Access and Insert/Update/Delete	DB2 Version 7.2 (11/12/01)
Time to Load, Backup, and Reconcile	DB2 Version 7.2 (03/10/02)

Results

Time to access a file on disk

Scenario

Measurements were performed with C applications using embedded SQL. The item table is fetched multiple times (up to 50 times). Tests were done with 100 rows of data. Each set of results gives the average time to access one file. Data Links and BLOBs are tested under the same conditions. Clients, the DB2 server, and the DLFM server are the only non-system processes working on the machines (except for the monitor process used to observe system activities).

Data Links — Time to get URL by select statement (files are already on disk). All DLFM processes (including the copy daemon) are idle before running any tests.
Configuration: Read Permission DB, Recovery yes, Write permission blocked, MAC0.

BLOB — Time to fetch BLOBs onto disk by select statement; the disk is located on the DLFM machine, same type as the DLFS file system disk. Configuration: Logged , Compact.

Single machine

8 clients saturate the CPU at about 98% with Data Links.
4 clients saturate the disk where files are fetched as BLOBs. The CPU is fully utilized on the client machine with 8 clients (waiting for disk I/O).

TPC-W standard database

DB2 V 7.2		1 client	4 clients	8 clients
Data Links	Images	1.4 ms	1.7 ms	2.1 ms
	Thumbnails	1.4 ms	1.7 ms	2.1 ms
BLOB	Images	41 ms	117 ms	196 ms
	Thumbnails	29 ms	78 ms	142 ms

Comments: Results are identical for data Links thumbnails and images, because only fetching of URLs is required (this is a major advantage of using Data Links over BLOBs).
Supplemental results: Data Links (50 clients): 14.9 ms. Access to files takes much longer with more clients, which demonstrates that the system is well saturated with 8 clients.

Big files database

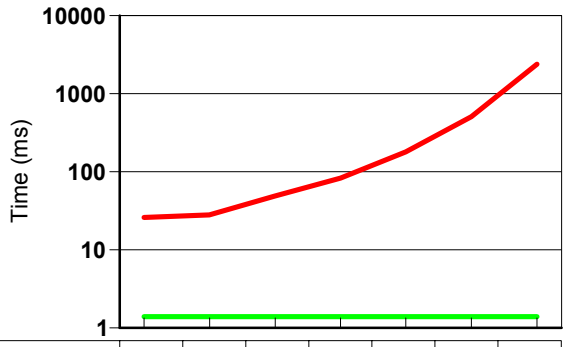
DB2 V 7.2	1 client	4 clients	8 clients
Data Links	1.4 ms	1.7 ms	2.1 ms
BLOB	1763 ms	4833 ms	9930 ms

Comments: This database has only one file column. Results are identical to that of the TPC-W database for Data Links. This table shows that file size doesn't matter for Data Links.

Time for several file sizes

Configuration: 1 client, only IMAGE column in item table is used for this test.
Note: Graph is logarithmic.

Figure6. *Time to access a file on disk on 1 machine*



	5 kb	10 kb	50 kb	100 kb	250 kb	1 Mb	5 Mb
BLOB	26	28	49	83	180	511	2386
Data Links	1.4	1.4	1.4	1.4	1.4	1.4	1.4

Comments: This table shows the obvious advantage of Data Links if you need to get a file on your file system.

Multiple machines

The TCP/IP communications protocol is used to access the remote node.
two clients saturate the CPU at 95% on the DB2 server (the Apollo computer is relatively slower) with Data Links. With 8 clients, the system is fully utilized. With BLOBs, data is sent through network to be written onto disk; thus network activity slows down the client and server machines. With 8 clients, the disk is practically saturated, at about 92%.

TPC-W standard database

DB2 V 7.2		1 client	2 clients	8 clients
Data Links	Images	3.5 ms	4.5 ms	14.4 ms
	Thumbnails	3.5 ms	4.5 ms	14.4 ms
BLOB	Images	87 ms	116 ms	283 ms
	Thumbnails	50 ms	64 ms	170 ms

Comments: Due to network activities, the network modulates overall system performance. For Data Links, the DB2 server is now on the Apollo computer (which is slower than the Nike computer).

Big files database

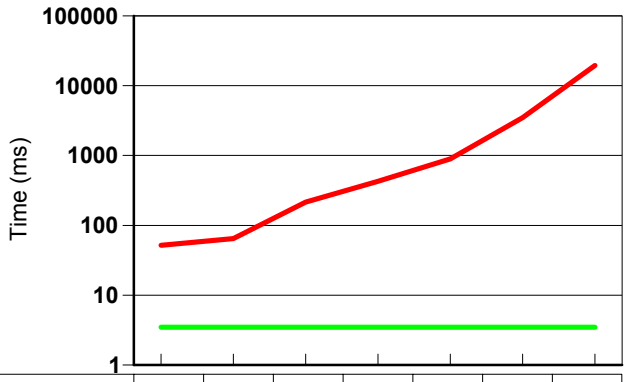
DB2 V 7.2	1 client	2 clients
Data Links	3.5 ms	4.5 ms
BLOB	4500 ms	6814 ms

Supplemental results: (8 clients, BLOB) 23000 ms.

Time with several file sizes

Configuration: 1 client, only IMAGE column in item table is used for this test.
Note: Logarithmic graph.

Figure 7. *Time to access a file on disk with multiple machines*



	5 kb	10 kb	50 kb	100 kb	250 kb	1 Mb	5 Mb
BLOB	52	65	217	431	903	3520	19530
Data Links	3.5	3.5	3.5	3.5	3.5	3.5	3.5

Comments: The need to avoid the movement of BLOBs is even more important with a network configuration.

Comparison Read permission DB / Read permission FS / Varchar (no link)

Configuration: 1 machine, TPC-W database, only image column is used.

DB2 V 7.2	1 client	8 clients
Read perm DB	1.4 ms	2.2 ms
Read perm FS	1.2 ms	2 ms
Varchar	1.2 ms	2 ms

Comments: Gives an idea for token generation time (comparing Read permission DB vs. Read permission FS).
Note: The VARCHAR column is assigned URL strings (just as with Data Links columns) to observe any difference in overhead for the Data Links column type. Comparison with the Data Links Read permission FS column shows virtually no additional overhead over the Varchar column type.

Comparison of MAC0 - MAC1

DB2 V 7.2	1 client	8 clients	50 clients
Mac0	1.44 ms	2.14 ms	14.9 ms
Mac1	1.57 ms	2.25 ms	14.9 ms

Comments: The Difference bettween MAC0 and MAC1 is quite small. Both give more or less the same results, especially with a heavier workload.

Time to load a file into memory

Scenario

Measurements were also performed with C-embedded SQL, and the same general conditions as accessing a file on disk. Results give the average time to access a file.

Data Links Time to get URL by SELECT statement + read file from DLFS into memory with standard C functions (fopen and fread). Files are loaded with 100 KB block size (different block sizes don't give significantly different results).
Configuration: Read Permission DB, Recovery yes, Write permission blocked, MAC0

BLOB Time to load a BLOB in memory by SELECT statement. BLOBs larger than or equal to 1MB are loaded with LOB locators using a 100 KB block size. Configuration: Logged, Compact.

Single machine

Each case is performed with 1 client and 10 clients. For Data Links or BLOBs, 10 clients enables the system to be saturated (the disk is saturated by multiple accesses).

TPC-W standard database

DB2 V 7.2		1 client	10 clients
Data Links	Images	12.5 ms	113 ms
	Thumbnails	11.5 ms	102 ms
BLOB	Images	12 ms	30 ms
	Thumbnails	9.8 ms	20 ms

Comments: BLOBs are faster with small files (see the chart below with different file sizes for more details).

Big files database

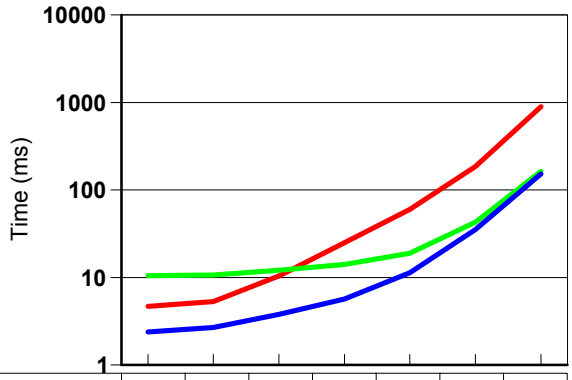
DB2 V 7.2	1 client	10 clients
Data Links	96 ms	174ms
BLOB	479 ms	1380 ms

Comments: The system is totally saturated for Data Links and BLOBs with 10 clients; this gives highest throughput when transferring from disk to local memory:
Data Links : 160 MB/s BLOBs : 20 MB/s
Note: The 10 clients access the same data so Data Links can take advantage of file system caching (more so than BLOBs).

Time with several file sizes:

Configuration: 1 client, only IMAGE column used for this test.
Note: Graph is logarithmic.

Figure 8. *Time to load a file in memory with one machine*



	5 kb	10 kb	50 kb	100 kb	250 kb	1 Mb	5 Mb
BLOB	4.7	5.3	10.4	25	60	185	896
Data Links Read DB	10.5	10.7	12.1	14.1	19	43	163
Data Links Read FS	2.4	2.7	3.8	5.7	11.3	35.4	153

Comments: From about 50 K, Data Links Read DB becomes faster than BLOBs to load the file into memory. With 5MB, it is five times faster.

Multiple machines

Network activities and separate machines allow the work load to be distributed across the resources on multiple machines. No single machine is saturated due to the network activities. Tests were done with 50 clients. The overall system throughput depends heavily on network activity. Therefore, results can be somewhat variable. The results here give average time over an extended period of testing.

TPC-W standard database

DB2 V 7.2		**1 client**	**10 clients**	**50 clients**
Data links	Images	34 ms	126 ms	681 ms
	Thumbnails	25 ms	91 ms	468 ms
BLOB	Images	24 ms	112 ms	425 ms
	Thumbnails	13 ms	32 ms	115 ms

Big files database

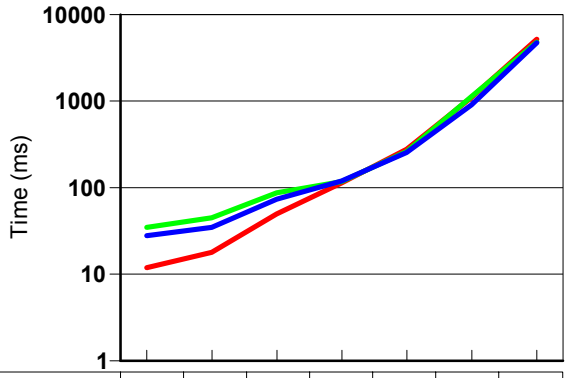
V 7.2	**1 client**	**10 clients**
Data Links	1.9 s	2.9 s
BLOB	1.8 s	12.7 s

Comments: The network is fully saturated. For any configuration, the network speed is about 10 Mbps. However with 10 clients, Data Links can take advantage of the NFS cache because the 10 clients access the same files.

Time with several file sizes

Configuration: 1 client, only image column used for this test. NFS cache not used (files are modified with the UNIX touch command before access in order to avoid caching).
Note: This graph is logarithmic .

Figure 9. *Time to load a file in memory with multiple machines*



	5 kb	10 kb	50 kb	100 kb	250 kb	1 Mb	5 Mb
BLOB	12	18	50	114	280	1112	5215
Data Links Read DB	35	45	88	118	263	1141	4821
Data Links Read FS	28	35	74	120	257	912	4748

Comments: BLOBs and Data Links show similar behavior. The network regulates the overall system speed (~1 MBps for both when network is saturated).
Supplemental results: With NFS, files can be in the NFS cache and so can be accessed very quickly. Blobs can never take advantage of the cache, because a separate file in the tablespace is used to store each BLOB. Here are some results with use of NFS caching.

(ms)	5k	10k	50k	100k	250k	1M	5M
Data Links Read perm DB	19.9	20.7	22.5	24.5	29.8	56	198
Data Links Read perm FS	6.2	7.2	8.7	11	16.3	42	179

The roughly fixed overhead of token generation and validation for Read permission DB versus Read permission FS is also easily observed here.

Comparison of Read permission DB / Read permission FS / Varchar:

Configuration: 1 machine, big files database, only image column used for this test.

DB2 V 7.2	1 client	10 clients
Read perm DB	96 ms	174 ms
Read perm FS	90 ms	137 ms
Varchar	90 ms	134 ms

Comments: Shows small overhead for token generation and validation between Read permission DB and FS.
Note: Results between MAC0 and MAC1 don't show a significant difference.

Time to insert/update/delete a file

Scenario

Measurements were performed using C applications with embedded SQL. For each database action, clients perform the same DML statement. Tests were done with 100 rows of data. Each result gives the average time to handle one file. For each case, the scenario is: drop table, create table, insert rows, update rows, delete rows. Tests were repeated several times to confirm results.

Data Links All files inserted and updated are in the DLFS file system (same machine as DLFM server). The archive disk and the dfm_file (file information) table in the DLFM_DB database are emptied before each test is started. Files inserted and updated are different. Update consists of linking a new file never before linked. Configuration: Read Permission DB, Recovery yes, Write permission blocked , MAC0.

BLOB All files inserted and updated are on a local disk where the DB2 server is located (as with the Data Links case). Configuration: Logged, Compact.

Important: When a row is inserted, all columns are inserted; when a row is updated, only the Data Links or BLOB column is updated. The TPC-W database has 2 file data columns (Image and Thumbnail) and the big files database has only one column (Image). Therefore, inserting a row in the tpcw database takes longer than inserting a row in the big files database with datalink columns. Results don't include the time to archive files.

Single machine

5 clients are allowed to saturate the DB2 server in both cases to 100%.

TPC-W Database (2 columns)

DB2 V 7.2	Statement	1 client	5 clients
Data Links	Insert	185 ms	422 ms
	Update	174 ms	362 ms
	Delete	115 ms	215 ms
BLOB	Insert	135 ms	270 ms
	Update	88 ms	160 ms
	Delete	38 ms	52 ms

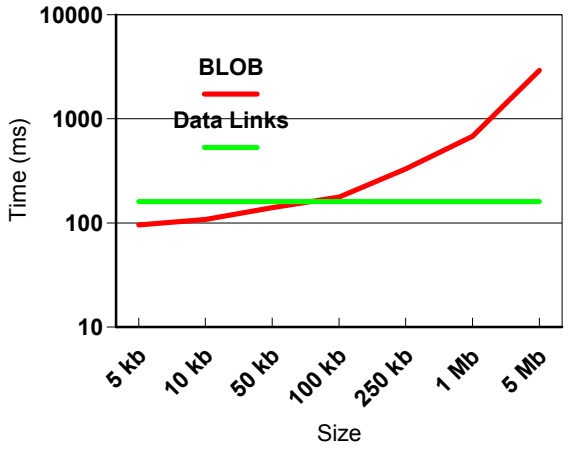
Big files Database (1 column)

DB2 V 7.2	Statement	1 client	5 clients
Data Links	Insert	161 ms	370 ms
	Update	143 ms	255 ms
	Delete	110 ms	195ms
BLOB	Insert	1300 ms	2500 ms
	Update	1000 ms	1600 ms
	Delete	38 ms	52 ms

Time with several file sizes

Configuration: 1 client, only image column in the item table is used for this test.
Only the chart for insert of a row is shown; update of a row gives approximately the same graphics; and time to delete is identical between BLOBs and Data Links for any file size.

Figure 10. *Time to insert a row with single machine*



Comments: Inserting a small BLOB is faster than inserting into a Data Links column, but with big files it is totally different.

Multiple machines

TPC-W database (2 columns)

DB2 V 7.2	Statement	1 client	5 clients
Data Links	Insert	143 ms	280 ms
	Update	132 ms	252 ms
	Delete	104 ms	157 ms
BLOB	Insert	206 ms	300 ms
	Update	130 ms	210 ms
	Delete	44 ms	57 ms

Comments: When files are inserted as blobs from clients, files are read into the client memory, and then inserted into the database. Data Links avoids this I/O because files are on the DLFM server before any insert and update. That is why Data Links is now faster with a network configuration.

Big files database (1 column)

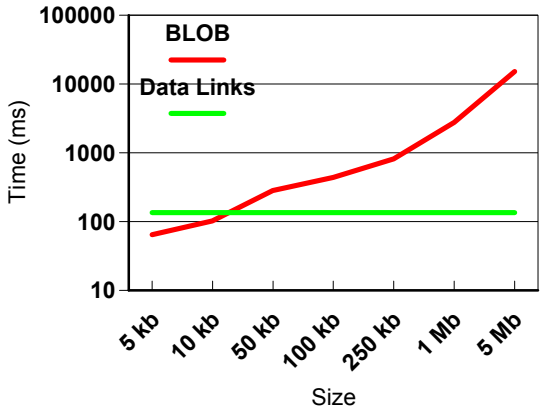
DB2 V 7.2	Statement	1 client	5 clients
Data Links	Insert	129 ms	253 ms
	Update	117 ms	182 ms
	Delete	95 ms	147 ms
BLOB	Insert	5400 ms	1520 ms
	Update	5900 ms	1580 ms
	Delete	45 ms	70 ms

Comments: With the Big files database, the time to handle blobs from the client becomes very long. See the chart below for more detail by file size.

Time with several file sizes

Configuration: 1 client, only IMAGE column used.

Figure 11 *Time to insert a row with multi machines*



Comments: In this configuration, it takes more than 15 seconds to insert a 5 MB blob from a remote client. Compare this to the eighth of a second for the Data Links value insert.

Comparison of Recovery no / Recovery yes / Read Permission FS:

Configuration: single machine, big files database (1 column), 1 client.
Option: Recovery no used to avoid copying linked file into the archive directory.

	Statement	DB2 V 7.2
Recovery yes - Read perm DB	Insert	161 ms
	Update	143 ms
	Delete	110 ms
Recovery no - Read perm DB	Insert	133 ms
	Update	126 ms
	Delete	113 ms
Recovery yes - Read perm FS	Insert	154 ms
	Update	124 ms
	Delete	110 ms

Comments: Time to insert and update is faster with recovery no. For delete, Data Links with Recovery no performs some additional synchronous update operations in the DLFM_DB (as opposed to Recovery yes, in which these updates are asynchronous, along with the file backup operation), and therefore takes a bit more time.

Time to back up

Scenario

For Data Links and BLOBs, the item table is loaded before running the backup process. Time to backup is measured with the UNIX 'time' command.
Configuration: Single machine, 1 client (running the backup process from a remote client doesn't produce different results).

Data Links Read Permission DB, Recovery yes, Write permission blocked , MAC0.

BLOB Logged, Compact

TPC-W database

1000 rows in the Tpcw database contains 19 MB.

DB2 V 7.2	100 rows	1000 rows	10000 rows
Data Links	27 s	29 s	30 s
BLOB	28 s	29 s	106 s

Big files database

100 rows in the big files database contains 2.8 GB.
Note: Backup for blobs is spanned across 4 directories by DB2.

DB2 V 7.2	100 rows	1000 rows
Data Links	28 s	32 s
BLOB	46 s	646s

Comments: Backup of blobs could be very long for large amounts of data. For Data Links, DB2 just needs to back up the URLs, and the files themselves are already backed up on the DLFM server.
Note: Use of Read Permission FS and Recovery no options don't affect results.

Time to load data

Scenario

The table is loaded from a 'del' file. Time is measured with the UNIX ‘time’ command. When the load or import command finishes loading data into the table, the DLFM copy daemon may still be copying files from the DLFS file system to the archive directory. The times shown below within parentheses give the total time to complete all processing including the Load and Copy processes. The Multiple Machines configuration doesn’t give different results, because all significant processing is performed on the DLFM server.

The following commands are used for the measurements:
- LOAD CLIENT FROM file OF DEL INSERT INTO ITEM
- IMPORT FROM file OF DEL INSERT ITEM

IMPORTANT: When the table is loaded, every column is loaded. The TPC-W database has an item table with 2 datalink columns, and the big files databases has 1 datalink column.

Configuration: Single machine, 1 client.

Data Links Read Permission DB, Recovery yes, Write permission blocked , MAC0.

TPC-W database (2 columns, 100 rows = 2 MB)

DB2 V7.2	100 rows	1000 rows	10000 rows
Load	6.8 s (18 s)	34 s (320 s)	348 s (5640 s)
Import	5.7 s (15 s)	46 s (157 s)	712 s (2586 s)

Comment: The import utility takes longer than the Load utility (for a larger number of rows). However, the time to copy files into the archive directory is less with Import (due to the steadier stream of activity to the DLFM compared to Load, which stresses the DLFM with large batches of work in periodic "chunks"). The archive disk activity is about 30%.

Big files database (1 column, 100 rows = 840 MB)

DB2 V7.2	100 rows	1000 rows
Load	4.9 s (51 s)	19 s (572 s)
Import	4.4 s (45 s)	36 s (469 s)

Comments: The archive disk activity is at about 95%, because files are bigger in this case.

Comparison with BLOBs

Configuration: Single machine, 1 client

TPC-W database (2 columns, 100 rows = 20 MB):

DB2 V7.2	100 rows	1000 rows	10000 rows
Load	6.5 s	62 s	1140 s

Big files database (1 column, 100 rows = 840MB)

DB2 V7.2	100 rows	1000 rows
Load	79 s	1096 s

Comments: This demonstrates the advantage of not using blob data (as opposed to datalinks) on the DB2 server.

Time to reconcile

The time to reconcile linked files is measured after a Load process . The UNIX 'time' command is used for measurements. In order to simulate a crash (loss of data) and cause Reconcile to restore lost files ("fix-up"), all files in the DLFS file system are first deleted. In this case, files are retrieved from the archive directory.

Data Links Read Permission DB, Recovery yes, Write permission blocked, MAC0.

Configuration : Single machine.

No fix-up required

TPC-W database (2 columns)

	100 rows	1000 rows
DB2 V7.2	7.5 s	15.2 s

Big files database (1 column)

	100 rows	1000 rows
DB2 V7.2	7.1 s	10.5 s

Fix-up required

TPC-W database (2 columns)

	100 rows	1000 rows
DB2 V7.2	106 s	994 s

Big files database (1 column)

	100 rows	1000 rows
DB2 V7.2	79 s	818 s

Note: Fix-up is not available with Recovery no (because files are not backed up to the archive directory). Running Reconcile from a remote client doesn't affect results, because most all processing is performed on the DLFM server.

Appendix

Database configuration

The following DB2 parameters were changed on the DB2 server:

- DL_NUM_COPIES=0
- DL_TIME_DROP=0
- MAXAPPLS=200

Item table With Datalink columns

TPC-W database

```
CREATE TABLE   "ITEM"  (  "I_ID" BIGINT NOT NULL ,   "I_A_ID" BIGINT NOT NULL ,
  "I_PUB_DATE" DATE NOT NULL , "I_RELATED1" BIGINT NOT NULL ,   "I_RELATED2"
BIGINT NOT NULL ,  "I_RELATED3" BIGINT NOT NULL ,   "I_RELATED4" BIGINT NOT
NULL ,  "I_RELATED5" BIGINT NOT NULL , "I_THUMBNAIL" datalink(200) linktype url
file link control read permission db write permission blocked integrity all
recovery yes on unlink restore,  "I_IMAGE" datalink(200) linktype url file link
control read permission db write permission blocked integrity all recovery yes
on unlink restore, "I_SRP" BIGINT NOT NULL ,"I_COST" BIGINT NOT NULL ,
"I_AVAIL" DATE NOT NULL ,"I_ISBN" CHAR(13) NOT NULL , "I_PAGE" SMALLINT NOT
NULL ,         "I_STOCK" SMALLINT NOT NULL ,  "I_TITLE" VARCHAR(60) NOT NULL ,
 "I_PUBLISHER" VARCHAR(60) NOT NULL ,   "I_SUBJECT" VARCHAR(60) NOT NULL ,
"I_DESC" VARCHAR(500) NOT NULL ,  "I_BACKING" VARCHAR(15) NOT NULL ,
"I_DIMENSIONS" VARCHAR(25) NOT NULL );
```

Big files database

```
CREATE TABLE   "ITEM"  (  "I_ID" BIGINT NOT NULL ,   "I_A_ID" BIGINT NOT NULL ,
  "I_PUB_DATE" DATE NOT NULL , "I_RELATED1" BIGINT NOT NULL ,   "I_RELATED2"
BIGINT NOT NULL ,  "I_RELATED3" BIGINT NOT NULL ,   "I_RELATED4" BIGINT NOT
NULL ,  "I_RELATED5" BIGINT NOT NULL , "I_THUMBNAIL" varchar(250) , "I_IMAGE"
datalink(200) linktype url file link control read permission db write
permission blocked integrity all recovery yes on unlink restore,"I_SRP" BIGINT
NOT NULL ,      "I_COST" BIGINT NOT NULL ,   "I_AVAIL" DATE NOT NULL ,
"I_ISBN" CHAR(13) NOT NULL ,        "I_PAGE" SMALLINT NOT NULL ,      "I_STOCK"
SMALLINT NOT NULL ,  "I_TITLE" VARCHAR(60) NOT NULL ,   "I_PUBLISHER"
VARCHAR(60) NOT NULL ,   "I_SUBJECT" VARCHAR(60) NOT NULL ,   "I_DESC"
VARCHAR(500) NOT NULL ,  "I_BACKING" VARCHAR(15) NOT NULL , "I_DIMENSIONS"
VARCHAR(25) NOT NULL );
```

Item table With Blob columns

TPC-W database

```
CREATE TABLE   "ITEM"  (  "I_ID" BIGINT NOT NULL ,   "I_A_ID" BIGINT NOT NULL ,
  "I_PUB_DATE" DATE NOT NULL , "I_RELATED1" BIGINT NOT NULL ,   "I_RELATED2"
BIGINT NOT NULL ,  "I_RELATED3" BIGINT NOT NULL ,   "I_RELATED4" BIGINT NOT
NULL ,  "I_RELATED5" BIGINT NOT NULL , "I_THUMBNAIL" BLOB(6k) LOGGED
COMPACT,"I_IMAGE" BLOB(300K) LOGGED COMPACT,"I_SRP" BIGINT NOT NULL ,
"I_COST" BIGINT NOT NULL ,   "I_AVAIL" DATE NOT NULL ,   "I_ISBN" CHAR(13) NOT
NULL ,         "I_PAGE" SMALLINT NOT NULL ,      "I_STOCK" SMALLINT NOT NULL ,
"I_TITLE" VARCHAR(60) NOT NULL ,   "I_PUBLISHER" VARCHAR(60) NOT NULL ,
"I_SUBJECT" VARCHAR(60) NOT NULL ,   "I_DESC" VARCHAR(500) NOT NULL ,
"I_BACKING" VARCHAR(15) NOT NULL , "I_DIMENSIONS" VARCHAR(25) NOT NULL );
```

Big files database

```
CREATE TABLE   "ITEM"  (  "I_ID" BIGINT NOT NULL ,   "I_A_ID" BIGINT NOT NULL ,
  "I_PUB_DATE" DATE NOT NULL , "I_RELATED1" BIGINT NOT NULL ,   "I_RELATED2"
BIGINT NOT NULL ,  "I_RELATED3" BIGINT NOT NULL ,   "I_RELATED4" BIGINT NOT
NULL ,  "I_RELATED5" BIGINT NOT NULL , "I_THUMBNAIL" varchar(250),"I_IMAGE"
BLOB(11M) LOGGED COMPACT,"I_SRP" BIGINT NOT NULL ,     "I_COST" BIGINT NOT NULL
,   "I_AVAIL" DATE NOT NULL ,   "I_ISBN" CHAR(13) NOT NULL ,         "I_PAGE"
SMALLINT NOT NULL ,   "I_STOCK" SMALLINT NOT NULL ,  "I_TITLE" VARCHAR(60) NOT
NULL ,   "I_PUBLISHER" VARCHAR(60) NOT NULL ,   "I_SUBJECT" VARCHAR(60) NOT
NULL ,   "I_DESC" VARCHAR(500) NOT NULL ,  "I_BACKING" VARCHAR(15) NOT NULL ,
"I_DIMENSIONS" VARCHAR(25) NOT NULL );
```

The table was loaded with the following command

```
LOAD CLIENT FROM <filename> OF DEL modified by lobsinfile INSERT INTO item;
```

Acknowledgements

The following people have contributed to this performance evaluation project and to this publication.

Laurent Perrimond	Primary tester, Document Author
Steven Elliot	Technical Leader, DB2 Data Links Development
Nhut Bui	DB2 UDB Performance
Berni Schiefer	Manager, DB2 UDB Performance
Frank Butt	DB2 Data Links Development
Suparna Bhattacharya	Architect, DB2 Data Links Development
Amit K Arora	DB2 Data Links Development
Terrie Jacopi	Manager, DB2 Data Links Development
James Yau	Manager, DB2 UDB Performance
Sunil Kamath	DB2 UDB Performance
Miso Cilimdzic	DB2 UDB Performance
George Ma	DB2 UDB Performance
Peter Shum	Manager, DB2 UDB Performance
Bill Burnett	System Administration Support
Elizabeth Wilde	Technical Editor

References

Data Links information	http://www-3.ibm.com/software/data/db2/datalinks/
DB2 Product documentation	http://www-3.ibm.com/software/data/db2/

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.